Złożoność obliczeniowa - ćwiczenia

1 Maszyny Turinga - podstawy

Jednotaśmowa, deterministyczna maszyna Turinga M ma następujące komponenty:

- alfabet wejściowy Σ,
- alfabet roboczy $\Gamma \supseteq \Sigma$ zawierający $\bot, \rhd \notin \Sigma$,
- zbiór stanów Q, oraz
- funkcja przejścia δ.

Maszyna dostaje na wejściu słowo x nad alfabetem wejściowym Σ, zapisane (prawie) na początku prawostronnie nieskończonej taśmy. Po lewej od słowa wejściowego x znajduje się niezmienialny znacznik $\rhd$, reszta taśmy jest wypełniona blankami $\bot$. Głowica jest na początku na $\rhd$, stan początkowy to jakiś ustalony q_0. Maszyna porusza się zgodnie z funkcją przejścia δ: w zależności od stanu i litery pod głowicą, maszyna zapisuje coś w komórce (lub nie, tzn. zapisuje to co tam było wcześniej), porusza się o jeden krok w lewo, w prawo, lub zostaje w miejscu, oraz zmienia stan. Zakładamy, że znacznika $\rhd$ nie da się nadpisać i nie można pójść na lewo od niego (zabronione w funkcji przejścia maszyny). Oprócz tego maszyna może przejść do specjalnego stanu q_A (Accept) lub q_R (Reject); wtedy obliczenie się kończy i słowo wejściowe jest akceptowane lub odrzucane.

Maszyna wielotaśmowa, jak sama nazwa wskazuje, ma wiele taśm (ustaloną z góry liczbę). Słowo wejściowe jest napisane na pierwszej taśmie, reszta jest wypełniona blankami. Na każdej taśmie jest głowica, przejścia zależą od stanu maszyny i krotki symboli pod głowicami, zaś instrukcje składają się z krotki instrukcji dla każdej taśmy (lewo/prawo/stój, oraz zapis).

Zadanie 1.1. Skonstruuj jednotaśmową maszynę rozpoznającą język palindromów w $O(n^2)$ krokach.

Zadanie 1.2. Skonstruuj dwutaśmową maszynę rozpoznającą język palindromów w $O(n)$ krokach.

Zadanie 1.3. Pokaż, że następujące modele zarówno dają się symulować przez jednotaśmową maszynę Turinga, oraz potrafią symulować jednotaśmową maszynę Turinga. W każdym przypadku oszacuj wzrost liczby kroków potrzebny do symulacji.

1. Maszyna jednotaśmowa, z obustronnie nieskończoną taśmą.
2. Maszyna wielotaśmowa, dla każdej ustalonej liczby taśm.
3. Maszyna jednotaśmowa, która ma co najwyżej 10 stanów.
4. Maszyna dwutaśmowa, która może pisać tylko po blankach.
5. Maszyna dwutaśmowa, w której pierwsza taśma z wejściem jest read-only, a druga, robocza, ma binarny alfabet roboczy.

Zadanie 1.4. Udowodnij, że jeśli istnieje wielotaśmowa maszyna rozpoznająca język L w co najwyżej $T(n)$ krokach, to również istnieje wielotaśmowa maszyna rozpoznająca L w co najwyżej $T(n)/2 + O(n)$ krokach.

Zadanie 1.5. Wykaż, że jeśli język L jest rozpoznawany przez maszynę jednotaśmową, która może pisać tylko po blankach, to L jest regularny.

Zadanie 1.6. Udowodnij, że jeśli język L jest rozpoznawany przez jednotaśmową maszynę M działającą zawsze w $O(n)$ krokach na wejściach długości n, to L jest regularny.

Język $L \subseteq \Sigma^*$ jest *częściowo obliczalny* jeśli istnieje maszyna, która akceptuje każde słowo z L, a na każdym słowie spoza L albo się pętli, albo je odrzuca. Język L jest *obliczalny* jeśli istnieje maszyna, która się zatrzymuje na każdym słowie wejściowym (ma własność stopu) i akceptuje słowa z L oraz odrzuca słowa spoza L.

Zadanie 1.7. Wykaż, że jeśli zarówno język L jak i jego dopełnienie $\overline{L} := \Sigma^* \setminus L$ są częściowo obliczalne, to L jest obliczalny.

2 Złożoność Kołmogorowa, złożoność komunikacyjna

Zadanie 2.1. Udowodnij, że jeśli $\alpha\colon \mathbb{N} \to \{0,1\}^*$ jest funkcją różnowartościową, to dla nieskończenie wielu wartości n zachodzi $|\alpha(n)| > \lfloor \log_2 n \rfloor$ (oznacza to, że żaden sposób zapisu liczb naturalnych nie może być bardziej efektywny niż tradycyjny zapis binarny).

Złożoność Kołmogorowa $K(w)$ słowa $w \in \{0,1\}^*$ to rozmiar najmniejszej maszyny Turinga M wypisującej w (gdy zostanie uruchomiona na pustym wejściu). Dokładniej, $K(w)$ to długość opisu maszyny M jako słowa nad alfabetem $\{0,1\}$.

Zadanie 2.2. Udowodnij, że istnieje nieskończenie wiele słów niekompresowalnych, tzn. takich, że $K(w) \geq |w|$.

Zadanie 2.3. Udowodnij, że funkcja K (złożoność Kołmogorowa) nie jest obliczalna.

Zadanie 2.4. Wykaż, że jeśli jednotaśmowa maszyna M rozpoznaje język palindromów binarnych, to maksymalna liczba kroków użyta przez M na wejściach długości n jest w $\Omega(n^2)$.

3 Klasa NP

Zadanie 3.1. Wykaż, że problem SAT dla formuł (niekoniecznie w postaci CNF), w których każda zmienna występuje co najwyżej dwa razy, jest NP-zupełny.

Zadanie 3.2. Udowodnij, że następujący problem kafelkowania jest NP-zupełny: dany jest zbiór kwadratowych kafelków (dla każdej uporządkowanej pary kafelków wiadomo, czy pasują do siebie w poziomie i w pionie), dwa wyróżnione kafelki t_1, t_2 oraz liczba n zapisana unarnie; pytamy, czy planszę $n \times n$ można wypełnić kafelkami (tak, aby wszystkie kafelki do siebie pasowały) kładąc t_1 w lewym dolnym rogu i t_2 w prawym górnym rogu.

Zadanie 3.3. Podaj przykład relacji wielomianowej R dla której oba zbiory $\exists R = \{x \mid \exists y.R(x,y)\}$ oraz $R\exists = \{y \mid \exists x.R(x,y)\}$ są NP-zupełne.

Zadanie 3.4. W problemie CLIQUE dany jest graf G i liczba k; pytamy się czy w G istnieje k parami sąsiadujących wierzchołków. Wykaż, że problem ten jest NP-zupełny.

Zadanie 3.5. W problemie SUBSET-SUM dana jest krotka liczb naturalnych $(a_1, \ldots, a_k, s)$ zapisanych w systemie binarnym i pytamy, czy dla pewnego podzbioru $I \subseteq \{1, \ldots, k\}$ zachodzi $s = \sum_{i \in I} a_i$. Udowodnij, że problem SUBSET-SUM jest NP-zupełny.

Zadanie 3.6. Problem UNARY-SUBSET-SUM różni sie od SUBSET-SUM tym, że liczy na wejściu zapisane są unarnie (liczbę n zapisujemy powtarzając n razy symbol 1). Udowodnij, że problem UNARY-SUBSET-SUM można rozwiązać w czasie wielomianowym.

Zadanie 3.7. W problemie PARTITION dana jest rodzina $\mathcal{X}$ podzbiorów zbioru $\{1, \ldots, n\}$ i pytamy, czy można wybrać podrodzinę $\mathcal{I} \subseteq \mathcal{X}$, która stanowi podział zbioru $\{1, \ldots, n\}$. Udowodnij, że problem PARTITION jest NP-zupełny.

Zadanie 3.8. Udowodnij, że jeśli P = NP to NEXPTIME = EXPTIME.

4 Klasa PSPACE

Zadanie 4.1. Udowodnij, że następujący problem kafelkowania jest PSPACE-zupełny: dany jest zbiór kwadratowych kafelków (dla każdej uporządkowanej pary kafelków wiadomo, czy pasują do siebie w poziomie i w pionie), dwa wyróżnione kafelki t_1, t_2 oraz liczba n_x zapisana unarnie; pytamy, czy istnieje n_y takie, że planszę $n_x \times n_y$ można wypełnić kafelkami (tak, aby wszystkie kafelki do siebie pasowały) kładąc t_1 w lewym dolnym rogu i t_2 w prawym górnym rogu.

Zadanie 4.2. Wykaż, że problem uniwersalności dla niedeterministycznych automatów na słowach jest PSPACE-zupełny.

Zadanie 4.3. Wykaż, że problem niepustości przecięcia dla automatów deterministycznych na słowach (dane automaty $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_m$; pytamy się czy $\bigcap_{i=1}^{m} L(\mathcal{A}_i) \neq \emptyset$) jest PSPACE-zupełny. A jak jest dla niedeterministycznych?

Zadanie 4.4. Rozważmy następującą grę, rozgrywającą się w danym grafie skierowanym G, przy ustalonym wierzchołku startowym v_0. Pionek początkowo stoi w v_0, a następnie gracze na przemian przesuwają pionek wzdłuż krawędzi, do wybranego przez siebie następnika obecnego wierzchołka, przy czym zabronione jest wejście do wierzchołka, w którym pionek już wcześniej stał. Przegrywa gracz, który nie może się ruszyć. Udowodnij, że problem „dane G i v_0, kto wygrywa?" jest PSPACE-zupełny.

Zadanie 4.5. Udowodnij, że klasy P, NP oraz PSPACE są zamknięte na gwiazdkę Kleene'ego, czyli jeśli L jest w klasie, to

$$L^* = \{u_1 u_2 \ldots u_n : u_i \in L\}$$

też jest w klasie.

5 Alternacja

Maszyna alternująca to maszyna Turinga o stanach podzielonych na $Q_\forall$ i $Q_\exists$. Przejścia albo prowadzą z jednej konfiguracji do drugiej, albo wywołują instrukcję akceptacji/odrzucenia. Maszyna akceptuje z konfiguracji ze stanem z $Q_\exists$ jeśli istnieje przejście, które albo natychmiast akceptuje, albo prowadzi do konfiguracji, z której maszyna akceptuje. Analogicznie dla konfiguracji ze stanami z $Q_\forall$ wymagamy by każde przejście miało tę własność. Maszyna działa w czasie $t(n)$ jeśli każdy bieg z konfiguracji początkowej dla słowa wielkości n ma długość co najwyżej $t(n)$ kroków. Maszyna używa pamięci $s(n)$ jeśli każdy bieg z konfiguracji początkowej dla słowa wielkości n zużywa co najwyżej $s(n)$ komórek pamięci roboczej. Klasy $\mathsf{ATIME}(t(n))$ oraz $\mathsf{ASPACE}(s(n))$ są zdefiniowane naturalnie. Oznaczamy $\mathsf{AL} = \mathsf{ASPACE}(\log n)$ oraz $\mathsf{AP} = \bigcup_{c\in\mathbb{N}} \mathsf{ATIME}(n^c)$.

Zadanie 5.1. Wykaż, że jeśli język L jest w klasie $\mathsf{ATIME}(t(n))$, to również dopełnienie L jest w tej klasie. Udowodnij również taki sam fakt dla klasy $\mathsf{ASPACE}(s(n))$.

Zadanie 5.2. Wykaż, że $\mathsf{AP} = \mathsf{PSPACE}$.

Zadanie 5.3. Wykaż, że $\mathsf{AL} = \mathsf{P}$.

6 Pamięć logarytmiczna (i mniejsza)

Definiując złożoność pamięciową maszyny nie wliczamy długości wejścia (które jest tylko do odczytu). Maszynę o tej własności nazywa się *maszyną wsadową*. Jeśli mówimy o maszynie obliczającej funkcję, to nie wliczamy także długości wyjścia (które jest tylko do zapisu – dopisujemy po jednej literce na końcu, nie możemy się cofać i czytać).

Zadanie 6.1. Wykaż, że język $\{a^k b^{2^k} \mid k \in \mathbb{N}\}$ da się rozpoznawać:
- W czasie $O(n \log n)$ na maszynie jednotaśmowej.
- W czasie $O(n)$ i pamięci $O(\log n)$ na maszynie wsadowej.

Zadanie 6.2. Na wejściu dany jest opis labiryntu: plansza to kwadrat $n\times n$ reprezentowany jako kolejne wiersze oddzielone #. Pola labiryntu to 0 (wolne pole), 1 (ściana), $\downarrow$ (wejście) oraz $\uparrow$ (wyjście). Zakładamy, że wejście i wyjście są na brzegu planszy, zaś pozostałe pola brzegu to ściany. Udowodnij, że da się rozstrzygnąć w pamięci logarytmicznej czy z wejścia da się dotrzeć do wyjścia. (Udowodnij, że język zawierający poprawne zapisy labiryntów, w których z wejścia da się dotrzeć do wyjścia, jest rozpoznawany w pamięci logarytmicznej).

Zadanie 6.3. Wykaż, że poprawne wyrażenia nawiasowe, przykładowo ((())()), da się rozpoznawać w logarytmicznej pamięci. A jak będzie jeśli w wyrażeniu moga się pojawiać dwa rodzaje nawiasów, () oraz []?

Zadanie 6.4. Dana jest formuła logiczna nie używająca zmiennych, innymi słowy, term nad operacjami logicznymi $\wedge$, $\vee$, $\neg$, gdzie w liściach są stałe. Wykaż, że można obliczyć wartość logiczną takiej formuły w logarytmicznej pamięci.

Zadanie 6.5. Ustalmy pewną skończoną algebrę $\mathbb{A}$: składa się ona ze skończonego nośnika A oraz skończonego zbioru funkcji $\mathcal{F}$. Każda funkcja $f \in \mathcal{F}$ ma ustaloną arność r, i działa z

A^r w A. Rozważmy następujący problem: dany term τ nad A ze stałymi w liściach, należy wyewaluować τ. Udowodnij, że ten problem da się rozwiązać w pamięci logarytmicznej.

Zadanie 6.6. *Automat wielogłowicowy* chodzi po słowie wejściowym przy pomocy pewnej stałej liczby głowic. Przejścia automatu zależą od krotki symboli pod głowicami, a operacja wykonana w przejściu polega na ruszeniu się każdej głowicy. Uwaga: automat nie pisze po słowie, jedynie czyta symbole i rusza głowicami. Wykaż, że dany język L jest rozpoznawalny przez pewien automat wielogłowicowy wtedy i tylko wtedy gdy jest rozpoznawalny przez maszynę Turinga działającą w logarytmicznej pamięci.

Zadanie 6.7. Język L jest zdefiniowany jako zbiór słów postaci:

$$\#0\ldots000\#0\ldots001\#0\ldots010\#0\ldots011\#\ldots\#1\ldots111\#$$

gdzie każdy blok ma długość k dla pewnego $k \in \mathbb{N}$ (mamy tutaj zapisy binarne wszystkich liczb od 0 do $2^k - 1$). Udowodnij, że L da się rozpoznać w pamięci $O(\log\log n)$. Uwaga: Należy zapewnić pamięć $O(\log\log n)$ także dla słów spoza języka.

Zadanie 6.8. Wykaż, że maszyna wsadowa o roboczej pamięci $O(1)$ może jedynie rozpoznawać języki regularne.

Zadanie 6.9. Rozważmy następujący problem obliczeniowy: mamy dany niedeterministyczny automat skończony A oraz słowo w (nad alfabetem wejściowym tego automatu). Pytamy się, czy A akceptuje w. Wykaż, że ten problem da się rozstrzygnąć w deterministycznej pamięci $O(\log|A| \cdot \log|w|)$.

Zadanie 6.10. Wykaż, że problem Directed Reachability jest zupełny dla NL (ze względu na redukcje w pamięci logarytmicznej) nawet jeśli założymy, że dany graf jest acykliczny i każdy wierzchołek ma całkowity stopień co najwyżej 3.

Zadanie 6.11. Wykaż, że $\mathtt{DIST} = \{(G, s, t, d) \mid$ odległość z s do t w G równa się $d\}$ jest NL-zupełny.

Zadanie 6.12. Wykaż, że klasa NL jest zamknięta na gwiazdkę Kleene'ego, natomiast klasa L jest zamknięta na gwiazdkę Kleene'ego wtw. gdy $\mathsf{L} = \mathsf{NL}$.

Zadanie 6.13. Wykaż, że klasa $\bigcup_{k\in\mathbb{N}} \mathsf{DSPACE}[\log^k n]$ nie ma problemu zupełnego (ze względu na redukcje w pamięci logarytmicznej).

7 Obwody logiczne

Zadanie 7.1. Wykaż, że dla każdego obwodu używającego bramek AND, OR, NOT istnieje równoważny obwód używający tylko bramek AND i OR, nie większej głębokości i mający co najwyżej dwukrotnie więcej bramek.

Zadanie 7.2. Funkcja $\mathsf{add}\colon \{0,1\}^{2n} \to \{0,1\}^{n+1}$ dodaje dwie liczby zapisane binarnie. Skonstruuj obwód dla add o wielomianowej liczbie bramek i stałej głębokości.

Zadanie 7.3. Funkcja majority dla danego ciągu bitów długości n zwraca 0 jeśli mniej niż połowa wejściowych bitów to jedynki, oraz 1 w przeciwnym przypadku. Skonstruuj obwód dla majority o liczbie bramek $O(n)$ i głębokości $O(\log n)$.

Zadanie 7.4. Skonstruuj obwód o wielomianowej wielkości i stałej głębokości dla funkcji parity, używający binarnych bramek AND, OR, NOT oraz bramek majority o nieograniczonym stopniu wejściowym.

Zadanie 7.5. Udowodnij, że każda funkcja $f\colon \{0,1\}^n \to \{0,1\}$ jest obliczalna przez obwód rozmiaru co najwyżej $1000 \cdot \frac{2^n}{n}$ (gdzie przez rozmiar rozumiemy liczbę krawędzi / drutów).

Zadanie 7.6. Niech p_n będzie prawdopodobieństwem, że losowo wybrana funkcja $f\colon \{0,1\}^n \to \{0,1\}$ jest obliczalna przez obwód rozmiaru co najwyżej $\frac{2^n}{10n}$ (gdzie przez rozmiar rozumiemy liczbę krawędzi / drutów). Udowodnij, że $\lim_{n\to\infty} p_n = 0$.

Zadanie 7.7. Powiemy, że język $L \subseteq \Sigma^*$ jest *rzadki* jeśli $|L \cap \Sigma^n| \leq p(n)$ dla pewnego wielomianu p. Udowodnij, że każdy język rzadki jest rozpoznawany przez pewien ciąg obwodów rozmiaru wielomianowego. Zauważ, że niektóre z tych języków są nierozstrzygalne.

Zadanie 7.8. Udowodnij, że język L należy do P/poly (czyli jest rozpoznawany przez ciąg obwodów rozmiaru wielomianowego, niekoniecznie jednorodny) wtedy i tylko wtedy, gdy istnieje rzadki język S taki, że L jest rozpoznawany w czasie wielomianowym przez maszynę z wyrocznią S.

8 Klasy AC/NC

Klasa AC^0 (odpowiednio, AC^1) obejmuje funkcje $\{0,1\}^* \to \{0,1\}$ obliczalne przez ciągi obwodów o wielomianowym rozmiarze i stałej głębokości (głębokości $O(\log n)$), w których bramki AND i OR mogą mieć dowolny stopień wejściowy. Klasa NC^1 obejmuje funkcje $\{0,1\}^* \to \{0,1\}$ obliczalne przez ciągi obwodów o wielomianowym rozmiarze i głębokości $O(\log n)$, w których bramki AND i OR mają stopień wejściowy co najwyżej 2. Klasy $\mathsf{u\text{-}AC}^0$, $\mathsf{u\text{-}NC}^1$ i AC^1 są zdefiniowane analogicznie, lecz dodatkowo wymagamy, aby istniała maszyna Turinga, która dla wejścia 1^n produkuje n-ty obwód z ciągu działając w pamięci logarytmicznej.

Zadanie 8.1. Wykaż, że każdy język regularny jest w $\mathsf{u\text{-}NC}^1$.

Zadanie 8.2. Język regularny jest *bezgwiazdkowy* (ang. *star-free*) jeśli można go opisać wyrażeniem regularnym nie używającym gwazdki Kleene'ego, tzn. opisywalnym gramatyką

$$R \to \alpha \mid \emptyset \mid (\neg R) \mid (RR) \mid (R + R)\,,$$

gdzie α to symbol alfabetu. Udowodnij, że każdy język bezgwiazdkowy jest w $\mathsf{u\text{-}AC}^0$.

Zadanie 8.3. Wykaż, że język opisów grafów spełniających ustaloną formułę logiki pierwszego rzędu jest w $\mathsf{u\text{-}AC}^0$.

Zadanie 8.4. Wykaż, że $\mathsf{u\text{-}NC}^1 \subseteq \mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{u\text{-}AC}^1$.

9 Algorytmy randomizowane

Klasa RP obejmuje języki, dla których istnieje wielomianowy algorytm randomizowany, który każde słowo z języka akceptuje z prawdopodobieństwem co najmniej 1/2, a każde słowo spoza języka zawsze odrzuca.

Klasa BPP obejmuje języki, dla których istnieje wielomianowy algorytm randomizowany, który każde słowo z języka akceptuje z prawdopodobieństwem co najmniej 3/4, a każde słowo spoza języka odrzuca z prawdopodobieństwem co najmniej 3/4.

Klasa PP obejmuje języki, dla których istnieje wielomianowy algorytm randomizowany, który każde słowo z języka akceptuje z prawdopodobieństwem co najmniej 1/2, a każde słowo spoza języka odrzuca z prawdopodobieństwem większym niż 1/2.

Klasa ZPP obejmuje języki posiadające wielomianowy algorytm Las Vegas: algorytm używający randomizacji, który jest zawsze poprawny, ale którego czas działania ma wartość oczekiwaną ograniczoną wielomianowo.

Zadanie 9.1. W problemie PERMUTATION k-PATH mamy dany graf skierowany G z wierzchołkami pokolorowanymi na k kolorów $1, 2, \ldots, k$. Pytamy się o istnienie ścieżki długości k takiej, że pierwszy wierzchołek jest koloru 1, drugi koloru 2, itd. Wykaż, że problem PERMUTATION k-PATH da się rozwiązać wielomianowo.

Zadanie 9.2. W problemie COLORFUL k-PATH mamy dany graf skierowany G z wierzchołkami pokolorowanymi na k kolorów $1, 2, \ldots, k$. Pytamy się o istnienie ścieżki długości k takiej, w której wierzchołki są parami różnych kolorów (ale niekoniecznie w określonej kolejności). Wykaż, że problem COLORFUL k-PATH da się rozwiązać w czasie $2^k \cdot poly(|G|)$.

Zadanie 9.3. W problemie k-PATH mamy dany graf skierowany G i pytamy się o istnienie ścieżki prostej na k wierzchołkach. Wykaż, że problem ten da się rozwiązać algorytmem Monte-Carlo z jednostronnym błędem (fałszywymi negatywami) w czasie $(2e)^k \cdot poly(n)$. Wysnuj wniosek, że pytanie o istnienie skierowanej ścieżki o długości $k = O(\log n)$ jest w klasie RP.

Zadanie 9.4. Udowodnij, że klasa BPP jest zamknięta na gwiazdkę Kleene'go.

Zadanie 9.5. Udowodnij, że jeśli $\mathsf{NP} \subseteq \mathsf{BPP}$ to $\mathsf{NP} = \mathsf{RP}$.

Zadanie 9.6. Udowodnij, że $\mathsf{ZPP} = \mathsf{RP} \cap \mathsf{coRP}$.

Zadanie 9.7. Udowodnij, że $\mathsf{NP} \subseteq \mathsf{PP}$.

Zadanie 9.8. Niech M będzie probabilistyczną maszyną Turinga, działającą w czasie wielomianowym i używającą logarytmicznej pamięci roboczej. Niech K będzie językiem zawierającym słowa w akceptowane przez M z prawdopodobieństwem co najmniej 1/2 (zatem słowa spoza K są odrzucane z prawdopodobieństem większym niż 1/2). Wykaż, że $K \in \mathsf{P}$.

10 Aproksymacja

Algorytm wielomianowy c-aproksymuje rozwiązanie problemu optymalizacyjnego minimalizacyjnego (maksymalizacyjnego), gdy dla każdego wejścia znajduje rozwiązanie o wartości co najwyżej (co najmniej) c-krotnie większe niż wartość rozwiązania optymalnego.

W problemie TSP dane są odległości $d_{ij} \geq 0$ między n miastami oraz ograniczenie B; pytamy, czy istnieje cykl przechodzący przez wszystkie miasta o długości co najwyżej B. W wersji optymalizacyjej, szukamy cyklu o najmniejszej długości. Na wykładzie pokazaliśmy, że jeśli $\mathsf{P} \neq \mathsf{NP}$, to w czasie wielomianowym nie da się c-aproksymować problemu TSP, dla dowolnej stałej c (a nawej dla dowolnej funkcji co najwyżej wykładniczej). W problemie MetricTSP odległości na wejściu spełniają dodatkowo nierówność trójkąta $d_{ij}+d_{jk} \geq d_{ik}$.

Zadanie 10.1. Udowodnij, że problem MetricTSP jest NP-zupełny.

Zadanie 10.2. Znajdź wielomianowy algorytm 2-aproksymujący dla problemu MetricTSP.

Zadanie 10.3. Znajdź wielomianowy algorytm 3/2-aproksymujący dla problemu MetricTSP.

W problemie MaxCnfSat mamy dany zbiór klauzul (formułę w postaci CNF) i chcemy znaleźć wartościowanie, które spełnia jak największą liczbę klauzul.

Zadanie 10.4. Znajdź wielomianowy algorytm 1/2-aproksymujący dla problemu MaxCnfSat.

Zadanie 10.5. Znajdź randomizowany wielomianowy algorytm 7/8-aproksymujący dla problemu MaxCnfSat działający przy założeniu, że w każdej klauzuli występują dokładnie 3 różne zmienne.

Zadanie 10.6. Znajdź (deterministyczny) wielomianowy algorytm 7/8-aproksymujący dla problemu MaxCnfSat działający przy założeniu, że w każdej klauzuli występują dokładnie 3 różne zmienne.